PathAppend

The output buffer must be sized to hold at least MAX_PATH characters

Sean Barnum, Cigital, Inc. [vita¹]

Copyright © 2007 Cigital, Inc.

2007-04-02

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 4685 bytes

Attack Category	Path spoofing or co	Path spoofing or confusion problem		
Vulnerability Category	Buffer Overflow			
	• Unconditional			
Software Context	File Path Managen	File Path Management		
Location	• shlwapi.h	• shlwapi.h		
Description	The output buffer for the PathAppend() function must be sized to hold at least MAX_PATH characters.			
	The PathAppend() function appends additional text to a path. The buffer used to return the path must be large enough to hold the returned value. The first parameter, pszPath, must be at least MAX_PATH characters in length to ensure that it is large enough to hold the returned string. Otherwise, a buffer overflow can occur.			
	Note: If the routine fails, it NULL's the path and returns FALSE. The routine will actually stop (and give FALSE) at the 260 MAX_PATH character limit.			
	Note: Some Unicode versions of Path functions can actually use paths that are up to 32,000 characters long by using a "\\?\" prefix on the path.			
APIs	Function Name	Comments		
	ATLPath::Append method	Overloaded wrapper of Path Append		
	PathAppend	arg 0+1 stored in 0		
	PathAppendA			
	PathAppendW			
Method of Attack	variable is not long enor Since the appendage dir	Attacker can cause a buffer overflow if the path variable is not long enough to hold the variable. Since the appendage directory can be of arbitrary length the attacker can overflow the buffer.		

^{1.} http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

PathAppend 1

Exception Criteria				
Solutions	Solution Applicability	Solution Description	Solution Efficacy	
	Whenever PathAppend is used.	Ensure that path buffer is at least MAX_PATH in length.	Effective.	
Signature Details	LPTSTR pszPath,	BOOL PathAppend(LPTSTR pszPath, LPCTSTR pszMore);		
Examples of Incorrect Code	TCHAR buffer \beta"); /, small! LPTSTR lpStr lpStr1 = but // String of TCHAR buffer TEXT("gamma" LPTSTR lpStr lpStr2 = but	<pre>LPTSTR lpStr1; lpStr1 = buffer_1; // String of what is being added. TCHAR buffer_2[] = TEXT("gamma"); LPTSTR lpStr2; lpStr2 = buffer_2;</pre>		
	<pre>bool ret = PathAppend(lpStr1,lpStr2);</pre>			
Examples of Corrected Code	TCHAR buffer TEXT(""alpha buffer is co	<pre>// String for path name. TCHAR buffer_1[MAX_PATH] = TEXT(""alpha\\beta"); // Note: buffer is correctly sized LPTSTR lpStr1; lpStr1 = buffer_1; // String of what is being added. TCHAR buffer_2[] = TEXT("gamma"); LPTSTR lpStr2; lpStr2 = buffer_2; bool ret =</pre>		
	TCHAR buffer TEXT("gamma LPTSTR lpStr lpStr2 = bui			
	PathAppend(lpStr1,lpStr2);			
Source Reference	url=/library/e	• http://msdn.microsoft.com/library/default.asp? url=/library/en-us/shellcc/platform/shell/ reference/shlwapi/path/pathappend.asp ²		
Recommended Resource				
Discriminant Set	Operating Syste	m • Wi	ndows	
	Languages	• C	+	

Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about "Fair Use," contact Cigital at copyright@cigital.com¹.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

PathAppend 3

^{1.} mailto:copyright@cigital.com